

근사 최적 이동 에이전트 이주 정책을 찾기 위한 클러스터 기반 방법

이재경, 김윤정, 배인한
대구효성가톨릭대학교 전자정보공학부

A Cluster-based Approach for Finding Near Optimal Agent Migration Strategies

Jae-Kyung Lee, Yoon-Jeong Kim, Ihn-Han Bae
Department of Computer Engineering
Catholic University of Taegu-Hyosung

요 약

이동 에이전트는 클라이언트 컴퓨터로부터 이주될 수 있고 실행을 위해 원격 호스트에 디스패치 될 수 있는 프로그램으로, 이동 가능하고 자율적이라는 속성 때문에 지난 몇 년 동안 많은 주목을 받아왔다. 하지만 이동 에이전트 기술이 상업적인 성공을 거두기 위해서는 에이전트 보안, 제어 구조, 트랜잭션 지원, 그리고 통신 모델의 개발과 같은 많은 기술적인 문제들이 해결되어야 한다. 본 논문에서는 에이전트가 통신 메카니즘으로 원격 프로시저 호출과 에이전트 이주를 함께 사용하는 경우, 이동 에이전트의 통신 성능을 향상시키기 위해 각 통신 순서마다 이동 에이전트가 이동할 수 있는 위치들의 집합을 동적으로 구성함으로써, 최적에 가까운 에이전트 이주 순서를 찾는 휴리스틱 알고리즘을 제안한다.

1. 서 론

최근 들어 폭넓게 분산된 이질적인 시스템을 위한 새로운 프로그래밍 패러다임으로 이동 에이전트 시스템이 많은 주목을 받고 있다. 이동 에이전트는 자신의 제어로 호스트 사이를 이주하고 각 호스트의 다른 에이전트와 상호 작용하거나 자원을 이용하여 맡겨진 임무를 수행하는 프로그램이다. 그 응용으로는 전자상거래, 그룹 공동 작업, 이벤트 모니터링, 작업 흐름 자동화, 정보 검색, 망 관리

및 이동 컴퓨팅이 있다.

분산 시스템을 위한 기존의 다른 방법들과 구별되는 가장 두드러진 특징은 에이전트의 이동성이라 할 수 있다. 에이전트 이주의 장점은 값비싼 전역 통신 비용 감소에 있다. 이 새로운 기술의 특징은 일단 네트워크가 사용자 컴퓨터와 서버간에 에이전트를 전송했다면, 그들 상호간에는 네트워크를 사용하지 않고 상호 작용할 수 있다. 즉, 어떤 진행 중인 상호 작용은 사용자와 서버 컴퓨터간에 계속적인 통신을 요구하지 않는다.

이동 에이전트의 이러한 이동가능하고 자율적인 속성은 분명, 분산 시스템을 위한 효율적인 패러다임이다. 하지만, 이동 에이전트 기술은 아직 초기 단계에 있으며, 상업적인 성공을 거두기 위해서는 에이전트 보안, 제어구조, 트랜잭션 지원, 그리고 통신 모델의 설계와 같은 많은 기술적인 문제들이 해결되어야 한다.

본 논문에서는 이동 에이전트가 통신 메카니즘으로 원격 프로시저 호출과 에이전트 이주를 함께 사용하고, 통신 순서가 고정되어 있는 경우, 그 에이전트의 통신 성능을 향상시키기 위해 클러스터에 기반한 근사 최적 에이전트 이주 정책을 제안한다.

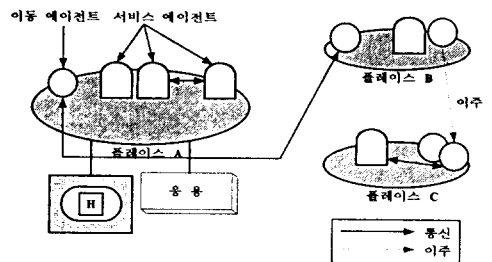
본 논문의 구성은 다음과 같다. 2장에서는 이동 에이전트 및 시스템 모델에 대해 설명하고, 3장에서는 에이전트 이주 정책과 관련한 기존의 연구를 살펴보고, 4장에서는 본 논문에서 제안하는 클러스터에 기반한 근사 최적 에이전트 이주 정책을 제안한다. 그리고 마지막으로 5장에서 결론을 맺는다.

2. 이동 에이전트 시스템

이동 에이전트 기술은 망 관리, 전자 상거래, 이동 컴퓨팅과 같은 분산 응용에 효율적인 패러다임이며, 네트워크에 일시적으로 연결되는 컴퓨팅을 지원하는데 효과적이다. 일시적으로 연결된 장치들은 SLIP나 PPP 모델 연결을 통해 네트워크에 가끔 연결되는 가정이나 사무실의 컴퓨터뿐 아니라, 랩탑이나 휴대용 네트워크 컴퓨터와 같은 물리적인 이동 컴퓨터를 포함한다. 이들 장치들은 긴 주기의 시간동안 네트워크로부터 종종 단절된다. 뿐만아니라, 비록 연결되었다 하더라도 고정 호스트에 비해 상대적으로 낮은 대역폭의 네트워크 연결과 제한된 저장용량 및 처리력을 가진다. 하지만, 이동 호스트가 갖는 이러한 문제는 이동 에이전트를 사용하여 쉽게 해결될 수 있다.

이동 호스트로부터 보내진 이동 에이전트

는 이동 호스트를 대신해서 정보를 모으기 위해 인터넷을 돌아다니고, 낮은 대역폭의 무선 연결을 통해 다수의 요청과 응답을 전송하는 대신 각 호스트의 자원을 이용하여 원하는 작업을 수행한다. 그리고 이동 호스트와 계속적으로 통신하지 않기 때문에, 비록 이동 호스트의 전력이 꺼지거나 네트워크로부터 단절되더라도 계속해서 자신의 작업을 수행할 수 있다. 사용자가 다시 연결되었을 때, 그 에이전트는 작업한 결과를 가지고 이동 호스트로 되돌아온다.



(그림 1) 에이전트 시스템 모델

에이전트 시스템 모델은 그림 1에서와 같이 에이전트와 플레이스로 구성되어진다. 에이전트 시스템은 에이전트들에게 다양한 서비스를 제공하는 많은 플레이스(Place)들로 구성된다. 플레이스는 이동 에이전트를 받아들이고, 수행 환경을 지원하며, 호스트 자원을 활용하여 다양한 서비스를 제공한다. 에이전트(Agent)는 크게 시스템 에이전트와 사용자 에이전트로 나누어진다. 시스템 에이전트는 특정 플레이스에 고정되어 있는 에이전트로, 다른 에이전트와 통신하거나 자신의 플레이스로 이주해 오는 이동 에이전트에게 서비스를 제공한다. 사용자 에이전트는 플레이스가 제공하는 서비스를 요청하기 위해 자신의 의지에 따라 플레이스간을 이주하는 이동 에이전트이다. 어떤 플레이스에서 다른 플레이스로의 여행(Travel)은 에이전트가 원격에서 제공되는 서비스를 사용할 수 있는 기능을 제공한다. 모임(Meeting)은 두 개 이상의 에이전트가 같은 플레이스에서 만나는

가능이다. 에이전트는 원하는 서비스를 제공하는 고정 에이전트나 또 다른 이동 에이전트와 만나기 위해 같은 플레이스로 여행할 수 있다.

이동 에이전트 기술의 두드러진 특징은 일단 네트워크가 사용자 컴퓨터와 서버간에 에이전트를 전송했다면, 그들 상호간에는 네트워크를 사용하지 않고 상호 작용할 수 있다는 것이다. 즉, 어떤 진행중인 상호 작용은 사용자와 서버 컴퓨터간에 계속적인 통신을 요구하지 않는다. 따라서 에이전트 이주는 RPC와 같은 다른 통신 메카니즘에 비해 값 비싼 전역 통신 비용을 줄일 수 있다는 장점이 있다. 하지만, 에이전트 이주와 원격 프로시저 호출의 특별한 교대 순서는 모든 가능한 플레이스에 대한 단순한 RPC 순서, 또는 단순한 에이전트 이주 순서보다 에이전트의 통신 성능을 향상시킨다[1].

3. 관련연구

M. Strasser의 연구[1]는 이동 에이전트 시스템에서 에이전트의 두가지 상호작용 모델: 원격 프로시저 호출과 에이전트 이주에 대한 성능 모델을 제안하였다. 여기서는 네트워크 부하와 실행 시간을 고려한 성능 모델이 개발되었고, 그 결과 이동 컴퓨팅에서 일반적인 시나리오에 대한 성능 모델의 응용은 에이전트의 최적 행위가 원격 프로시저 호출과 에이전트 이주의 혼합된 순서에 의해 성취되어짐을 보였다.

Iqbal의 연구[2]는 이동 에이전트의 통신 순서가 고정된 경우, 최적 결정 그래프를 이용하여 최적 에이전트 이주 정책을 위한 정확한 알고리즘을 제시하였다. 여기서 에이전트는 고정된 순서로 다수의 플레이스와 통신한다. 에이전트가 통신하는 플레이어의 집합을 SCP로 나타내고, 에이전트가 이동할 수 있는 위치들의 집합을 SMP로 표현한다. 그리고 에이전트는 CS로 알려진 고정 순서로 플레이스들과 통신한다. 다른 플레이어와 통

신하기 위하여 에이전트는 이주하고 그 이주 순서는 MS로 표현된다. 즉, 이동 에이전트는 CS의 순서로 SCP에 속하는 플레이어들과 통신하며, 그 플레이어들과 통신하기 위해서 MS의 순으로 SMP내의 위치들 사이를 이동한다. 이동 에이전트 통신은 에이전트가 MS(i-1)에서 MS(i)로 이주하고, 다시 MS(i)에서 CS(i)로의 원격 프로시저 호출을 통해 이루어진다. 위치 a에서 위치 b로의 에이전트 이주 비용을 MCost(a,b)라 하고, 위치 c에서 위치 d로의 원격 프로시저 호출 비용을 RCost(c,d)라고 할 때, i번째 통신을 만들기 위한 비용은 다음과 같다.

$$Cost(i) = MCost(MS(i-1), MS(i)) + RCost((MS(i), CS(i))$$

따라서, 전체 통신 비용은 $\sum_{i=1}^n Cost(i)$ 이며, 에이전트의 이주는 이 비용이 최소가 될 수 있도록 이루어져야 한다. 여기서, n 은 에이전트와 플레이어간의 통신회수를 나타낸다.

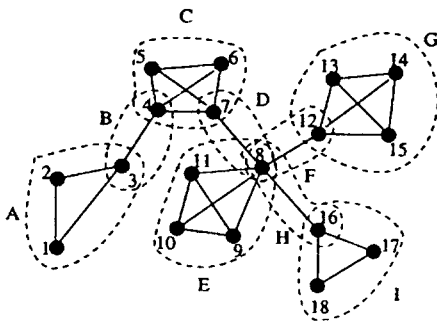
최적 에이전트 이주 순서는 최적 결정 그래프를 이용하여 얻을 수 있다. 최적 결정 그래프는 통신 순서 크기 만큼의 많은 계층을 갖는 계층 그래프이고, 추가적으로 시작 노드와 끝 노드를 포함한다. 시작 노드와 끝 노드는 각각 에이전트의 시작 위치와 최종 목적지를 나타낸다. 결국, 최적 이동 에이전트 이주 순서는 최적 결정 그래프의 시작과 끝 노드간의 최단 경로를 구함으로써 얻어진다. 하지만, 최적 이주 경로를 찾기 위해서는 최적 결정 그래프의 각 계층의 모든 정점들이 에이전트가 이동할 수 있는 위치들의 집합(SMP)을 모두 고려해야 하기 때문에 상당히 많은 계산 시간을 요구할 뿐만 아니라, 인터넷위킹과 같은 환경에서 최적 에이전트 이주 순서를 찾는다는 것은 사실상 불가능하다. 따라서, 본 논문에서는 [2]에서처럼, RPC와 에이전트 이주를 수행할 때의 통신 비용이 실행전에 미리 알려져 있고 이동 에이전트의 통신 순서가 고정된 경우, 에이전트의 통신 순서에 따라 SMP를 동적으로 구

성하기 위해 클러스터에 기반한 근사 최적 에이전트 이주 정책을 제안한다.

4. 클러스터 기반 에이전트 이주 정책

4.1 개요

본 논문에서는 RPC 비용과 에이전트 이주 비용은 실행전에 미리 알려져 있고, 이동 에이전트의 통신 순서는 미리 고정되어 있다고 가정한다. 이동 에이전트의 근사 최적 이주 경로를 찾기 위해서 [3]의 방법을 사용하여 전체 네트워크 노드들이 최소한 1개의 클러스터내에 반드시 포함되도록 전체 네트워크를 여러개의 클러스터로 나눈다(그림 2).



(그림 2) 네트워크 모델

이 때, k-클러스터를 어떤 고정된 값 k에 대해, 최소 k의 경로 길이에 의해 서로 도달할 수 있는 노드들의 부분집합이라고 정의할 때, 클러스터의 크기는 이 k값에 의해 달라질 수 있다. 그리고 네트워크내의 각 노드들은 클러스터 리스트와 바운드 리스트를 유지한다. 클러스터 리스트는 클러스터들마다 각 클러스터를 구성하고 있는 노드들의 집합을 나타내고, 바운드 리스트는 서로 겹쳐지는 클러스터들간의 경계 노드를 나타낸다.

본 논문에서 각 클러스터들은 클러스터내의 노드들간에 경로 길이가 1인 1-클러스터이고, 계층 구조를 갖지 않는 단일-계층 클러스터를 고려한다. 이 경우에 클러스터 리스트 구조와 바운드 리스트 구조는 표1, 표2와 같다.

<표 1> 1-클러스터 리스트

ClusterID	Nodes
A	1,2,3
B	3,4
C	4,5,6,7
D	7,8
E	8,9,10,11
F	8,12
G	12,13,14,15
H	8,16
I	16,17,18

<표 2> 바운드 리스트

ClusterID	Node
A,B	3
B,C	4
C,D	7
D,E,F,H	8
F,G	12
H,I	16

이동 에이전트의 최적 이주 경로를 결정하기 위해서 이들 리스트를 참고하여 이주 결정 그래프의 각 계층별로 SMP를 동적으로 구성한다. 통신 순서 CS(i)에서는 CS(i)와 CS(i+1)을 포함하는 클러스터들의 노드들로 이주 결정 그래프의 SMP(i)를 구성한다. 그리고, 이 그래프의 시작 노드에서부터 끝노드까지의 최단 경로 길이를 구함으로써 근사 최적 에이전트 이주 경로를 얻을 수 있다.

4.2 알고리즘

클러스터에 기반한 근사 최적 에이전트 이주 경로를 찾기 위한 알고리즘은 그림 3과 같다.

Algorithm: Find Near Optimal Agent Migration Sequence

Input: Cluster_List, Bound_list, CS(i), 1 ≤ i ≤ z;

MCost(i,j), RCost(i,j), 1 ≤ i,j ≤ z;

Output: MS(i), 1 ≤ i ≤ z;

Cluster ← ∅;

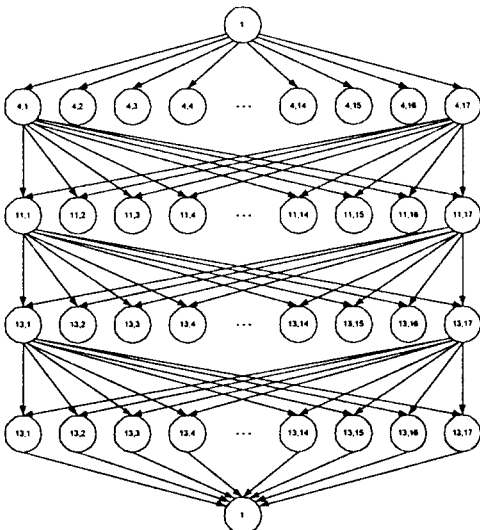
Place the layer 0 of MDG(migration decision graph) in start vertex(S);

```

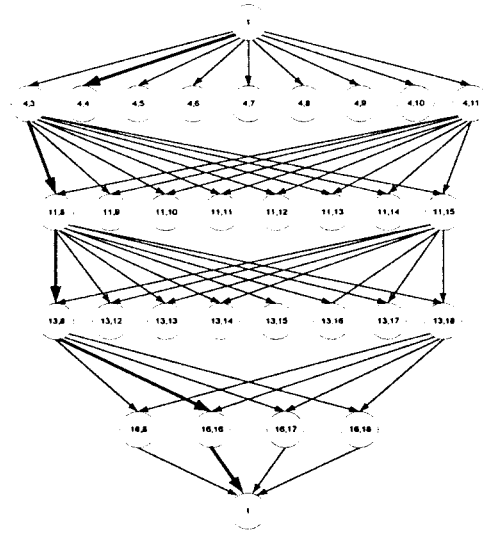
For i=1 to z do
  For j=i to i+1 do
    If CS(j) ∈ Bound_List.Node then
      Cluster ← Bound_List.ClusterIds;
    Else
      Cluster ← Cluster_List.ClusterId;
    Endif
  Endfor
  For all cluster k ∈ Cluster do
    Place the layer i of MDG in the vertices as the size of
    cluster k, these vertices are labelled as cd, where c=CS(i)
    and d ∈ the nodes in the cluster k;
    Each vertex ab in layer i-1 is connected to the vertex cd
    in layer i, the cost associated with the edge(ab, cd) is
    Cost(ab, cd) = MCost(b,d) + RCost(d,c);
  Endfor
Endfor
Each vertex ab belonging to the zth layer is connected to the end
vertex(E), the cost associated with edge (ab, E) is
MCost(b, home);
Find the shortest path between the start to the end vertex
in the MDG using Dijkstra's algorithm;
    
```

(그림 3) 에이전트 이주 순서를 찾기 위한 클러스터 기반 알고리즘

예를 들어, 그림 2의 네트워크 모델에서 이동 에이전트의 홈 플레이스가 1이고, CS={4, 11, 13, 16}이라고 할 때, Iqbal의 최적 결정 그래프는 그림 4와 같고, 본 논문의 이주 결정 그래프는 그림 5와 같다.



(그림 4) Iqbal의 최적 결정 그래프



(그림 5) 이주 결정 그래프

4.3 시간 복잡도

본 논문에서 제안하는 클러스터 기반 에이전트 이주 알고리즘의 시간 복잡도는 다음과 같다.

【정리】 본 논문에서 제안하는 알고리즘의 시간 복잡도는 $O(n^2)$ 이다.

【증명】 n개의 노드들로 구성된 네트워크 시스템, 에이전트는 CS에 따라 z번 플레이스들과 통신한다고 가정하자. 본 논문의 알고리즘에 의해 계층 i에 생성되는 정점들의 개수(m)은 다음과 같다.

$$m = | \text{Member}(\text{cluster}(CS(i))) \cup \text{Member}(\text{cluster}(CS(i+1))) | \leq n$$

여기서 Member는 클러스터를 구성하는 멤버 노드들을 반환하는 함수이다. 계층 i의 모든 정점들은 계층 i+1의 모든 정점들과 연결되고, MDG내의 계층의 개수는 z이므로 본 논문에서 제안하는 알고리즘의 시간 복잡도는 다음과 같다.

$$zm^2 = \Theta(m^2) \leq O(n^2) \quad \square$$

5. 결론

본 논문에서는 에이전트 이주와 원격 프로

시저 호출을 함께 사용하는 경우, 통신 성능을 향상시키기 위한 이동 에이전트의 이주 경로를 찾는 알고리즘을 제안했다. 이 알고리즘은 전체 네트워크를 다수의 클러스터로 나누고, 통신 순서의 각 단계마다 SMP를 동적으로 구성하여 이주 결정 그래프를 작성하였다. 그런 다음, 이 그래프의 출발노드에서 종료노드까지 최단 경로를 이동 에이전트의 이주 순서로 정함으로써, 이동 에이전트의 근사 최적 이주 경로를 구할 수 있다. 앞으로의 연구과제는 본 논문에서 제안한 방법을 시뮬레이션을 통해 성능을 분석하고 평가하는 것이다.

Science, Dartmouth College, May, 1996.

[6] 김평준, 윤석환, 이동 에이전트를 위한 이동 에이전트 시스템, 정보처리학회지 Vol. 4 No. 5, pp.67~75, 1997.

참고문헌

- [1] Markus Strasser, Markus Schwehm, A Performance Model for Mobile Agent Systems, Proceeding of the International Conference on Parallel and Distributed Processing Techniques and Applications PDPTA'97, 1997.
- [2] M. Ashraf Iqbal, Joachim Baumann, Efficient Algorithms to Optimal Agent migration Strategies, Technical Report TR-1998-05, University of Stuttgart, 1998.
- [3] P. Krishna, M. Chatterjee, N. H. Vaidya, and D. K. Pradhan, "A Cluster-based Approach for routing in Dynamic Networks", ACM SIGCOMM Computer Communications Review(CCR), 1997.
- [4] J. Baumann, F. Hohl, M. Strasser, K. Rothermel, Mol-Concepts of a Mobile Agent System, accepted for www Journal, Special issue on Applications and Techniques of Web Agents, Baltzer Science Publishers, 1998.
- [5] Robert Gray, David Kotz, Saurab Nog, Daniela Rus, George Cybenko, Mobile Agents for Mobile Computing, Technical Report PCS-TR96-285, Dept. of Computer